CPU **130** folds the side panels through the fold angles. If graphics engine **138** is used, the transformation of the side panels is performed by graphics engine **138**. Alternatively, CPU **130** just calculates new corners of the side panels and their projections onto the view plane. Once the side panels are in place, CPU **130** proceeds to block **306**.

At block **306**, the four corner panels are moved to rejoin the side panels. When the side panels are folded, the corner panels are unattached to their adjacent side panels except at one point where the corner panel touches its two adjacent side panels and the focus panel. Since the angle of the corner panel at a touching corner is a right angle and the angle between the edges of the previously adjacent two side panels is no longer a right angle, the corner panel is sheared so that the touching corner has an angle equal to the angle formed by the edges of the two adjacent side panels. An example of this is shown in FIG. **5**, where side panel edges **222** and **224** which define angle A at touching point **220**.

Once CPU **130** shears the corner panel, CPU **130** the rotates the corner panel, and then folds it down to rejoin the two adjacent side panels. Of course, in some embodiments, the shearing, rotation and folding steps might be performed in a different order. Depending on the particular fold angles of its adjacent side panels, the corner panel might be folded once to rejoin with one side panel, and then folded along the rejoined edge to rejoin the other side panel edge.

With the side panels continuous with the focus panel after folding and the corner panels continuous with the side panels after folding, a continuous folded surface in 3D containing the layout of the full image within the solid angle of the display surface is created.

At block **307**, the 3D folded surface embodying the full image is projected onto the view plane. Because the side panels and corner panels have been folded at an angle to the z=0 plane, their perspective projections will be smaller than if they were viewed head on, and the detail of the image get smaller further from the focus panel.

At block **308**, CPU **130** renders each of the focus, side and corner panels. A panel is rendered by identifying the bounds of the panel's projection onto the display surface, identifying the pixels of the display device which are within those bounds, identifying the bounds of the panel on the original full image, mapping the pixels of the display device onto rendering points of the original full image using the transformations calculated for the panel, determining pixel colors for each rendering point on the original full image, and then assigning those pixel colors to pixels of the display device. Because of the placement of the focus panel and the display surface in the 3D space, the transformations for the focus panel will result in a denser set of rendering points on the focus panel of the full image than the density of rendering points near the far edges of the side or corner panels (which are farther away from the viewpoint in the 3D space). Consequently, the focus panel will appear in finer detail, and since the panels are flat and continuous in the 3D space, context information is preserved. Because the panels are flat, the computation of the transforms is simple and the resulting image remains discernable. For improved rendering speed, where the image is text, the text within the side and corner panels could be greeked, either all the time or just during image movements. If greeking is used, the text in the focus panels is rendered in the font for that text, but the text within the side and corner panels is rendered in a greeked font. In a greeked font, a line of text is rendered as a single line.

If the objects of the full image which call on specific panels have been identified, then all objects which don't appear in whole or part on a given panel can be removed from the full image when that panel is being rendered.

After rendering the image, in some embodiments, CPU **130** perform one or more optional steps, shown by blocks **309**. **310** and **311**. At block **309**, the side and corner panels are shaded to direct the viewer's attention to the focus panel while still keeping image portions on the side and corner panels visible. At block **310**, CPU **130** adds border lines around the panels if desired. Alternatively, CPU **130** only adds border lines around the focus panel.

At block **311**, if frame buffers are used, they are swapped here. Frame buffers are used where manipulating the image as it is being displayed would be undesirable. In such as case, the image is rendered to one frame buffer which is not the frame buffer being displayed, that frame buffer is swapped to become the current display. Another frame buffer is then used form rendering of subsequent images.

At block **312**, CPU **130** exits the program if some predetermined exit condition is met, otherwise it proceeds to block **313**. At block **313**, CPU **130** checks for a movement command, either moving the image relative to the focus panel (a pan), or moving the viewpoint in and out (a zoom), and in embodiments where it is allowed, CPU **130** checks for a viewpoint movement command. If a movement command has not been entered, CPU **130** moves back to block **312**, otherwise it loops back to block **300** to create a new display based on the movement command.

FIGS. **7** and **8** are examples of substitute images which are perspective sheets. The full image in FIG. **7** is a uniform grid with one sentence "Read me if you can." roughly in the center of the image. FIG. **7** illustrates, with its grid, how the image is shaped over each panel. The full image in FIG. **8** is a 17-page document in a six by three page array. In rendering the display of FIG. **8**, as explained above in block **308** (FIG. **6**), a simplified image having only one page is used for rendering the focus panel, while the left side panel can use a simplified image of three pages. The rendering for the other panels is similarly simplified.

The above description is illustrative and not restrictive. Many variations of the invention will become apparent to those of skill in the art upon review of this disclosure.

For example, a "display device" or display system, as used herein, could be a computer monitor, printer, or the like, although computer monitors are more suitable for interactive image display. A display system might also include an image processor or other computer system for specialized processing of an image before display. A display device might also be a device which behaves as if it displayed images but does not actually display the images. For example, a facsimile machine might manipulate images much the same manner as a computer display, but doesn't necessarily display an image, instead providing the image to another system which does display the provided image.

The scope of the invention should, therefore, be determined not with reference to the above description, but instead should be determined with reference to the appended claims along with their full scope of equivalents.

An appendix, listing source code for a specific embodiment of a perspective sheet display system is included in the application.

What is claimed is:

1. A method for presenting an interactive navigation of an image where context and detail information are preserved even when the image contains too much detail to be displayed all at once on a display surface, the method comprising the steps of: